



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

M.S. Dissertation of Engineering

**Enhancing Attribute-Factorized  
Representations in Variational  
Autoencoder by Regularizing  
Multiple Mutual Information  
Elements**

August 2019

Graduate School of  
Convergence Science and Technology  
Seoul National University  
Program in Digital Contents and Information

Sedong Kim

**Enhancing Attribute-Factorized Representations  
in Variational Autoencoder by Regularizing  
Multiple Mutual Information Elements**

지도교수 이 원 중

이 논문을 공학석사 학위논문으로 제출함

2019년 8월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

김 세 동

김세동의 공학석사 학위论문을 인준함

2019년 8월

위 원 장	<u>서 봉 원</u>	(인)
부 위 원 장	<u>이 원 중</u>	(인)
위 원	<u>이 교 구</u>	(인)

## Abstract

# Enhancing Attribute-Factorized Representations in Variational Autoencoder by Regularizing Multiple Mutual Information Elements

Sedong Kim

School of Convergence Science & Technology

The Graduate School

Seoul National University

Recently, there have been many studies on deep generative models that can learn representations of data and generate new samples. We consider learning representations of target attributes and representations of the other attributes and how to disentangle them in deep generative models, and introduce a new Variational AutoEncoder (VAE) based generative model named as MMVAE (Multiple Mutual information elements VAE). The objective function of MMVAE can enhance attribute-factorized representations by regularizing multiple mutual information elements. Specifically, we construct a framework that explicitly regularizes mutual information of each pair among attributes,

attribute representations, and the other representations by adopting Mutual Information Neural Estimation (MINE, Belghazi et al., 2018). In the model, the objective function consists of an evidence lower bound and three mutual information regularizers. The formulation corresponds to a minimax game, where a group of parameters in autoencoder is optimized to minimize the objective function while another group in mutual information regularizers is optimized to maximize the objective function. We demonstrate, through a series of experiments on CelebA datasets, that the model can learn the target attribute representations and the other representations in better factorized forms and that these factorized representations are useful for generating images with the target attributes.

**Keywords:** deep generative model, representation learning, attribute factorization, regularizing multiple mutual information elements

**Student Number:** 2017-21650

# Table of Contents

<b>I. Introduction</b>	1
<b>II. Related Works</b>	8
2.1 VAE and CVAE	8
2.2 Recent Works for Attribute-Factorization	9
2.3 Mutual Information Neural Estimation	15
<b>III. Research Questions and the Proposed Method</b>	17
3.1 Research Questions	17
3.2 Model	18
3.2.1 ELBO	18
3.2.2 Mutual Information Regularization	21
3.2.3 Objective Function	23
3.3 Implementation	24
3.4 Evaluation methods	27
<b>IV. Experimental Results</b>	29
4.1 Experimental Setup	29
4.1.1 Dataset	29
4.1.2 Architecture of Neural Networks and Training	30
4.2 Experimental Results	33
4.2.1 Qualitative Results	33
4.2.2 Quantitative Results	36

<b>V. Conclusion</b> .....	42
5.1 Conclusions .....	42
5.2 Contributions .....	43
5.3 Limitations.....	43
 <b>References</b> .....	46

## List of Tables

Table 1.	Summarization of generative models using attribute labels .....	13
Table 2.	Mutual information on each model.....	40
Table 3.	Accuracy of classification of reconstructed images .....	41

## List of Algorithms

Algorithm 1.	Bayesian Optimization .....	26
--------------	-----------------------------	----

## List of Figures

Figure 1.	Structure of variational autoencoder.....	2
Figure 2.	Manifold of MNIST learned by VAE .....	3
Figure 3.	Relationship between entropy and mutual information.....	6
Figure 4.	The encoder and decoder for VAE and CVAE.....	9
Figure 5.	Structure of models in previous studies.....	11
Figure 6.	Structure of MMVAE .....	19
Figure 7.	Graphical model of true generation .....	19
Figure 8.	The two dimensional attribute representation space.....	32
Figure 9.	Generated images on CelebA-Glasses by each model.....	34
Figure 10.	Attribute changed images on CelebA by MMVAE.....	35
Figure 11.	Mutual information depending on the hyperparameters .....	37
Figure 12.	Trends of loss and mutual information values of CVAE .....	38
Figure 13.	Trends of mutual information values of CSVAE & MMVAE ....	39



# Chapter 1. Introduction

Recently, artificial intelligence (AI) has attracted considerable attention due to its diverse application areas and an infinite amount of potential. In the early days of artificial intelligence, researches were restricted to simple models, such as linear regressions, logistic regressions, support vector machines, and decision trees. Using the simple models, artificial intelligence cannot solve complex problems containing extremely tangled pattern structures in a data space of an amount of dimensions. These problems include tasks that are relatively easy and intuitive for human beings, such as recognition objects in images and understanding context in sentences. But, thanks to the success of deep neural networks, artificial intelligence started to solve the complex problems using a hierarchy concept which mimics an operation of a human brain. Now, many people look forward to artificial intelligence which can diagnose disease in medicine, diagnose failure in industry, translate foreign language, analyze scientific data, and interpret sentences or videos.

One of the most interesting and challengeable research area of artificial intelligence is to construct a deep generative model which can generate new data samples that the model have never learned before. The deep generative models are expected to draw images using desired objects and style, generate text while keeping a consistent subject and

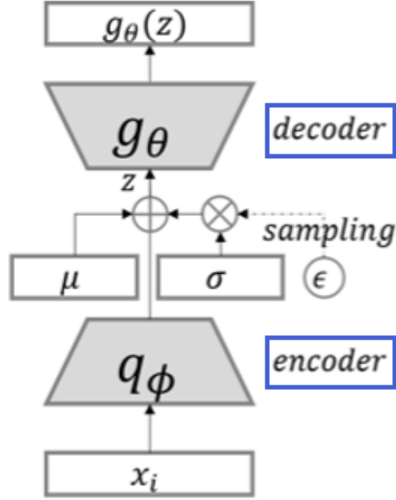


Figure 1. Structure of variational autoencoder. It can learn bi-directional mapping between a data space and representation space by its encoder and decoder.

literary style, and compose music using desired mood and rhythm on behalf of human beings in the near future. The most successful deep generative models are variational autoencoder (VAE, Kingma & Welling, 2013), generative adversarial net (GAN, Goodfellow et al., 2014), and their variants. Such models learn a mapping from a latent representation space to a data space, such as natural language and images, in the decoder of VAE or the generator of GAN. Especially, VAE and its variants also learn a counter-directional mapping from a data space to a latent representation space by the encoder as shown in Figure 1. Thus VAE-based models can be used to manipulate the shape of the latent representation space (Makhzani et al., 2015), or edit specific data

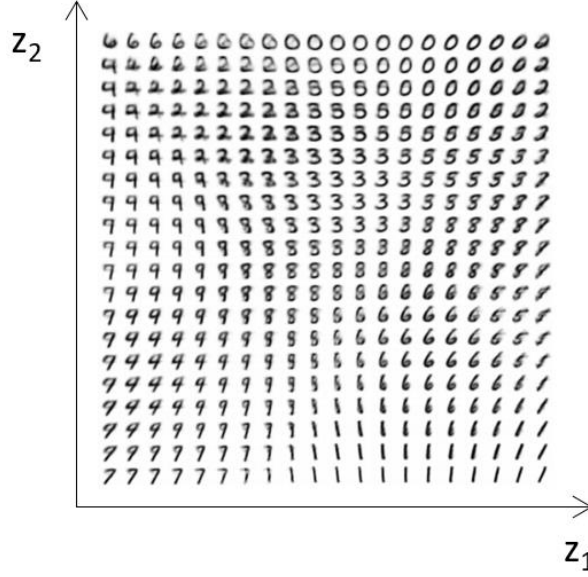


Figure 2. Manifold of MNIST learned by VAE (Kingma & Welling, 2013). The compact and smooth representation is spread in a latent space.

example by adjusting its representations in the latent space (Lample et al., 2017).

VAE learns the compact and smooth representation in the latent space. Figure 2 represent the smooth change of representation in the latent space. Thus VAE-based models can generate various kinds of samples by varying latent variables for representation in the latent space. A small change of semantically specific meaningful attributes in the data space, for example the change of tilting angle in a digit-image, is well matched to a small change toward a specific direction of latent value in the latent space. However in the plain VAE, we cannot force the specific

latent variables to learn desired attributes. Matching between latent variables and desired attributes is simply determined by an uncontrolled process during the training and initial random setting of model parameters. In order to overcome this limitation, conditional variational autoencoder (CVAE, Sohn et al., 2015) uses true label data of the attributes. Then, the model can learn a mapping from the true label and latent representation space to the data space in a supervised manner. In other words, CVAE can conditionally generate samples using the desired attribute information.

Although CVAE-based models have achieved impressive progress generating attribute-conditional images, the latent space including the attribute information is not entirely separated from the latent space including the other representation information. Consequently, CVAE cannot preserve the attributes when it generates samples by changing latent variables of the other representations (Bao et al., 2017). For example, when CVAE generates images of a man wearing eyeglasses (the desired attribute is eyeglasses), it could be difficult to preserve the eyeglasses when we change the other representations including hair color, skin tone and age, etc.

One of the key challenges in the recent deep generative models is how to factorize the desired attribute information and the other representations information of data examples into different latent spaces in a disjoint way. We call the representation learned in latent spaces by the disjoint property *attribute-factorized representation*. If a deep generative model learns the attribute-factorized representation, the

model can manipulate desired attribute representations and the other representations independently. And then, the model can also generate data samples which contain the desired attributes and preserve the other representations and vice versa. To achieve learning the attribute-factorized representation in the encoder of VAE, there have been several researches based on CVAE including Fader Networks (Lample et al., 2017), IFcVAE-GAN (Creswell et al., 2017), and CSVAE (Klys et al., 2018). We will introduce these models in section 2.2. Unlike these models, our model (MMVAE: multiple mutual information VAE) explicitly regularizes multiple mutual information elements during the training process, between each pair among true attribute labels, attribute representations, and the other representations.

In information theory, mutual information is a powerful measure of the dependence between two random variables. It quantifies the total amount of information obtained from one random variable when another random variable is observed. Figure 3 shows the relationship between entropy and mutual information of two random variables. Notice that the mutual information  $I(X;Y)$  corresponds to the intersection of the information in  $X$  with the information in  $Y$  (Cover & Thomas, 2012). The mutual information  $I(X;Y)$  between two continuous random variables with joint density  $p(x,y)$  is defined as

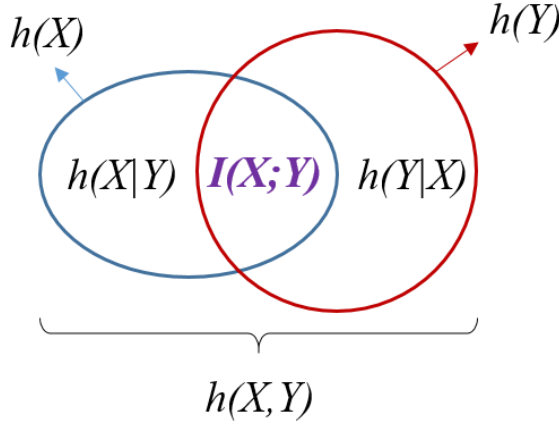


Figure 3. Venn diagram of Relationship between entropy and mutual information.

$$\begin{aligned}
 I(X; Y) &= h(X) - h(X|Y) = h(Y) - h(Y|X) \\
 &= h(X) + h(Y) - h(X, Y) \\
 &= \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy.
 \end{aligned}
 \tag{1-1}$$

However, this integral is usually intractable except for special cases such as Gaussian joint density. Also, almost always there is no way to know the probability distributions of representation variables in a neural network. There are several methods to approximate the mutual information using the empirical distribution related to data examples, such as a binning method (Shwartz-Ziv & Tishby, 2017), a k-nearest neighbors based model (Kraskov et al., 2004), and a kernel density estimator based model (Kolchinsky et al, 2017). The state of the art is

Mutual Information Neural Estimation (MINE, Belghazi et al., 2018). This method offers the tight lower bound of the true mutual information by using neural network estimator. MINE will be discussed in more detail in section 2.3. We employ MINE to approximate the mutual information in MMVAE.

We analyze and compare the previous works in chapter 2. Next we propose research questions, and introduce MMVAE and methods evaluating the models in chapter 3. We show and discuss qualitative and quantitative results of experiments in chapter 4. In the last chapter, we conclude our studies and discuss the limitations of this study.

## Chapter 2. Related Works

In this chapter, we first describe variational autoencoder (VAE, Kingma & Welling, 2013) and conditional variational autoencoder (CVAE, Sohn et al., 2015) which form the basis of our model. Next, we describe the recently developed models for factorizing the desired attribute information from the other representation information, and then compare these models. And finally, we introduce Mutual Information Neural Estimation (MINE, Belghazi et al., 2018) that is used for approximating the mutual information.

### 2.1 VAE and CVAE

Each of VAE and CVAE consist of an encoder and a decoder that use neural networks. Unlike VAE, CVAE uses variable  $y$ , which is true attributes or class labels, as inputs for both encoder and decoder as shown in Figure 4. Thus CVAE can learn a mapping from a latent representation space with specified attributes to a data space and also can generate samples with the desired attributes. Both models optimize the tractable evidence lower bound (ELBO) instead of maximizing intractable marginal log likelihood. Then their loss functions are the negative ELBO, which becomes the upper bound of negative marginal log likelihood. The loss functions for VAE and CVAE are represented as the followings,



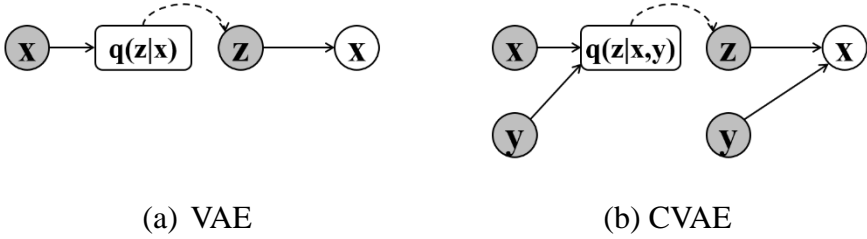


Figure 4. Encoder and decoder for VAE and CVAE. The variables  $x$ ,  $y$ , and  $z$  are the data example, the true attribute label, and the representation, respectively. Shaded nodes indicate conditioning variables. Dashed lines represent i.i.d. sampling.

respectively.

$$\begin{aligned}
 -\log p_{\theta}(x) &\geq -\mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] \\
 &\quad + D_{KL}\left(q_{\phi}(z|x)||p(z)\right),
 \end{aligned} \tag{2-1}$$

$$\begin{aligned}
 -\log p_{\theta}(x|y) &\geq -\mathbb{E}_{q_{\phi}(z|x,y)}[\log p_{\theta}(x|z,y)] \\
 &\quad + D_{KL}\left(q_{\phi}(z|x,y)||p(z)\right)
 \end{aligned} \tag{2-2}$$

The first term of the loss function for each model means reconstruction error, while the second term (the Kullback-Leibler (KL) divergence) acts as a regularizer which make a latent space dense.

## 2.2 Recent Works for Attribute-Factorization

In this section, we briefly analyze and compare several deep

generative models that use attribute information to generate desired images. Figure 5 shows illustrations of these models. All of them use attribute label ( $y$ ) and additional neural networks to regularize dependency between an attribute and other nodes as shown in red boxes.

Fader Networks (Lample et al., 2017) uses a discriminator to classify attributes ( $y$ ) from representation ( $z$ ), as a regularizer in its loss function. The discriminator is trained to answer the correct attribute, while encoder ( $E_\phi$ ) is trained to deceive the discriminator as the method in GAN (Goodfellow et al., 2014). Then, to make a mapping to representation, encoder uses only the information except the attribute information that data example ( $x$ ) originally includes. In this case, representation tend to be independent of the attribute.

CVAE-GAN (Bao et al., 2017) uses a similar discriminator to classify attributes ( $y$ ) from reconstructed samples ( $\hat{x}$ ). In that case reconstructed samples can contain the attribute information more confidently, but representation cannot well separate from the attribute information, namely, encoder ( $E_\phi$ ) cannot learn attribute-factorized representation.

IFcVAE-GAN (Creswell et al., 2017) use both discriminators in Fader Networks and CVAE-GAN. Thus representation ( $z$ ) can be independent of the attribute ( $y$ ), and reconstructed samples can contain the attribute information.

But, these three models use the true attribute label ( $y$ ) directly instead of an additional attribute representation variable as the input to the decoder. Generally, labels included in a dataset are not continuous

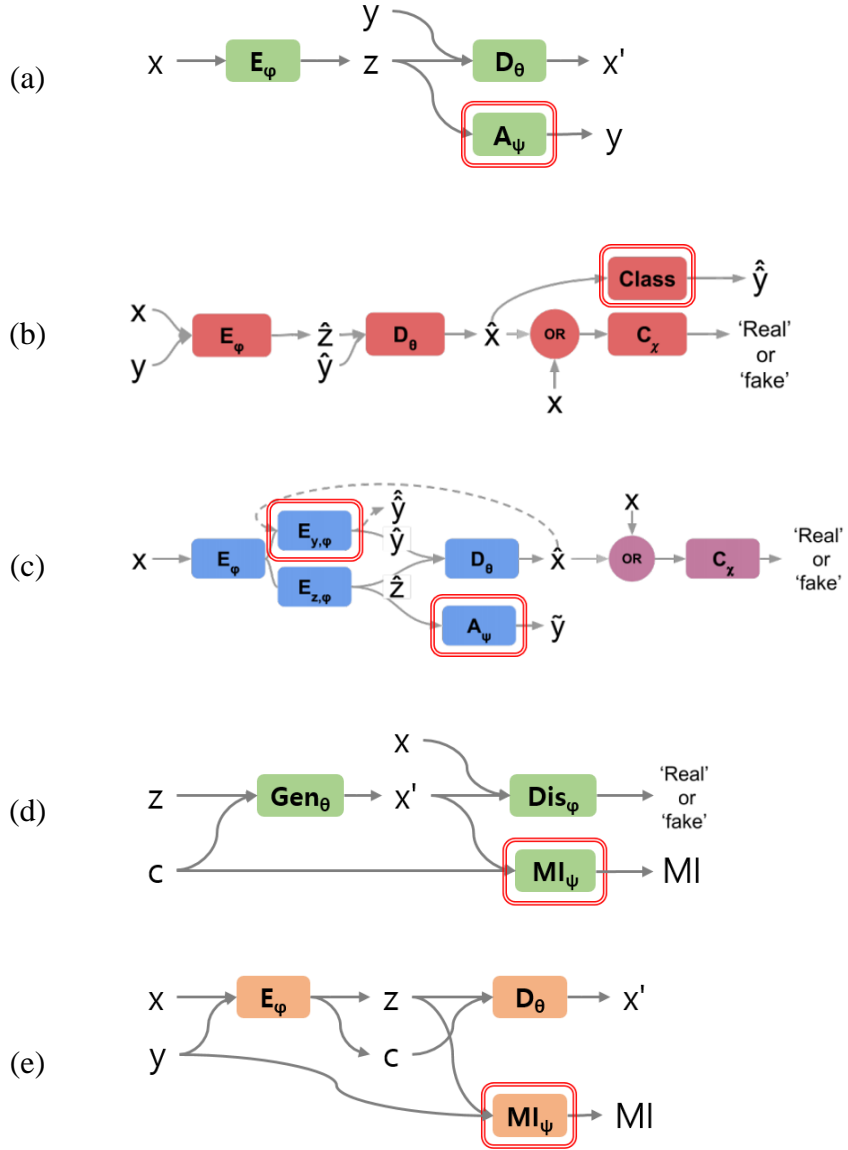


Figure 5. Structure of each model. (a) Fader Networks, (b) CVAE-GAN, (c) IFcVAE-GAN, (d) InfoGAN, (e) CSVAE. (b) and (c) are from (Creswell et al., 2017).

variables but discrete variables, usually binary. The labels explain simple high-level information (ex. wearing eyeglasses), but not details (color of lens or thickness of the frame) contained in data examples ( $x$ ). Hence it is superior to add an attribute representation variable to learn the detailed features related to the binary labels in data examples (Klys et al., 2018).

InfoGAN (Chen et al., 2016) uses an additional attribute representation variable ( $c$ ), whose mutual information with reconstructed samples ( $x'$ ) is aimed to be maximized. The mutual information is approximated using pre-set  $p(c)$  and an approximation of  $p(c|x')$  via an auxiliary neural network ( $MI_{\psi}$ ). In the model, the prior distributions of attribute representation variable ( $c$ ) and the variable for the other representations ( $z$ ) are pre-determined by the user. As a result, those two variables can be definitely independent. However, we cannot choose which of attributes are learned in attribute representation variable ( $c$ ) due to the absence of an encoder.

CSVAE (Klys et al., 2018) also use an additional attribute representation variable ( $c$ ), which has minimized mutual information with the other representation variable ( $z$ ). However it cannot manipulate dependency between the attribute representation and the other representations. Hence the attribute representation variable can erroneously contain the other representation information even though the other representation variable can be independent of true attribute ( $y$ ). This model approximates mutual information using pre-set  $p(y)$  and calculated  $p(c|x')$  from a neural network ( $MI_{\psi}$ ).

Table 1. Summarization of generative models using attribute labels

Model	Dependency regularization	Missing for attribute-factorization	Encoder used?	Attribute label used?	Attribute representation used?
Fader Net	min. dep(z, y)	-	O	O	<b>X</b>
CVAE-GAN	Max. dep(x', y)	<b>min. dep(z, y)</b>	O	O	<b>X</b>
IFcVAE-GAN	Max. dep(x', y) min. dep(z, y)	-	O	O	<b>X</b>
InfoGAN	Max. dep(x', c)	<b>min. dep(z, c)</b>	<b>X</b>	<b>X</b>	O
CSVAE	min. dep(z, y)	<b>min. dep(z, c)</b> <b>Max. dep(y, c)</b>	O	O	O

In both models of InfoGAN and CSVAE, mutual information approximators use the pre-set prior distribution,  $p(c)$  or  $p(y)$ . Provided that such a prior distribution is not given, the approximator cannot estimate mutual information because it is intractable to calculate the marginal distribution. For this reason, the mutual information between attribute representation variable (c) and the other representation variable (z) in CSVAE cannot be calculated using this approximator. We will discuss this in more detail in section 3.2.2.

We summarize the properties of each model in Table 1. The red and bold letters indicate their weaknesses that are insufficient to learn attribute-factorized representation.

Unlike the above-mentioned models, there are several generative models that manipulate attribute representations without using the true labels of the attribute in an unsupervised manner. One of the mainstream is style-transfer and cross-domain image generation based unsupervised GAN (Taigman et al., 2016, Li et al., 2016, Huang et al., 2017, Karras et al., 2019). Especially, Karras et al. proposed a novel style-based generator architecture that can selectively transfer various levels of aspects or features as well as the overall style of the image (Karras et al., 2019). But, these models cannot manipulate a given specific attribute in contrast to the supervised ones.

Another stream aims to learn structured and disentangled representations in VAE-based model (Higgins et al., 2017, Kim et al., 2018, Chen et al., 2018, Esmaili et al., 2019). They decompose the KL-divergence terms in evidence lower bound (ELBO) of VAE and regularize it in various ways in training, so that the various information on the images is disentangled in the representation space. After training, several axes in the representation space can represent independent attributes regardless of whether the attributes are included in the label or not. Chen et al. (2018) reported that 15 attributes had been discovered among 40 kinds of labels on CelebA dataset by their model without any supervision.

## 2.3 Mutual Information Neural Estimation

Mutual Information Neural Estimation (MINE) is based on Donsker-Varadhan representation (Donsker & Varadhan, 1983) of KL divergence represented as

$$\begin{aligned} D_{KL}(\mathbb{P}||\mathbb{Q}) &= \sup_{T:\Omega \rightarrow \mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T]) \\ &\geq \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}}[T_{\theta}] - \log(\mathbb{E}_{\mathbb{Q}}[e^{T_{\theta}}]), \end{aligned} \tag{2-3}$$

where the first supremum is taken over all functions  $T$  and second supremum is taken over a subset of functions possible to be represented as  $\theta$  such that each expectation is finite.

Mutual information is equivalent to the Kullback-Leibler (KL) divergence between the joint and the product of the marginals of two random variables. Therefor the lower bound of the mutual information can be represented as

$$\begin{aligned} I(x; z) &= D_{KL}(\mathbb{P}_{xz}||\mathbb{P}_x \otimes \mathbb{P}_z) \\ &\geq \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}_{xz}}[T_{\theta}] - \log(\mathbb{E}_{\mathbb{P}_x \otimes \mathbb{P}_z}[e^{T_{\theta}}]) = I_{\Theta}(x; z). \end{aligned} \tag{2-4}$$

$T_{\theta}$  can be constructed using sufficiently large neural networks and proper training. Then,  $I_{\Theta}$  becomes a very tight lower bound of true MI by universal approximation theorem (Hornik et al., 1989; Cybenko, 1989;

Leshno et al., 1993) of neural networks.

In the previous section, InfoGAN and CSVAE regularize dependency between two variables by introducing mutual information term as a regularizer in their loss function. But they cannot handle the mutual information between any variables whose prior distributions are unknown. To solve the problem, we employ MINE in our VAE-based generative model to calculate a tight lower bound of the true mutual information using only empirical distributions even if the prior distributions are unknown. Then we can explicitly regularize the mutual information because MINE can calculate a tight lower bound regardless of the types of variables.



## **Chapter 3. Research Questions and the Proposed Method**

Considering previous works, we first present research questions for our studies in this chapter. Next, we discuss how we build and materialize our model to answer the research questions. Methods for evaluating conditional generative models are introduced as well in this chapter.

### **3.1 Research Questions**

This study aims to construct a new model and loss function to enhance independent relation between attribute representations and the other representations, namely to learn attribute-factorized representations in VAE-based generative models using the concept of mutual information. We consider the following two research questions.

- RQ 1. How should the model and loss function be enhanced to better learn attribute-factorized representations in VAE-based generative models?
- RQ2. What is a proper strategy to train VAE-based generative models while simultaneously approximating the mutual

information between two variables whose marginal distributions are unknown?

## 3.2 Model

We have constructed MMVAE to learn attribute-factorized representation by regularizing mutual information among variables of a true attribute ( $y$ ) and latent variables ( $z, c$ ) as shown in Figure 6. There are three pairs of variables whose the mutual information is regularized in MMVAE. First, we maximize  $I(c; y)$  in order to make  $c$  contain all the information related to the desired attributes. The supremum  $I(c; y)$  is equal to  $H(y)$ , which is easily calculated from empirical distribution of attribute labels. Second,  $I(z; y)$  is minimized so that  $z$  never includes the information related to the desired attributes. Third,  $I(z; c)$  is also minimized to make  $z$  and  $c$  independent. We use MINE method to calculate the tight lower bound of the three mutual information values. In this section, we show the objective function for MMVAE.

### 3.2.1 ELBO

First, we introduce the basic evidence lower bound for our VAE-based model except mutual information terms. This basic ELBO is the same as that of CSVAE (Klys et al., 2018).

Let  $\mathcal{X}$  be a data domain and  $\mathcal{Y}$  be a domain of attributes assigned to data in  $\mathcal{X}$ . Here we consider binary attribute  $y \in \mathcal{Y} = \{0, 1\}^k$  and its

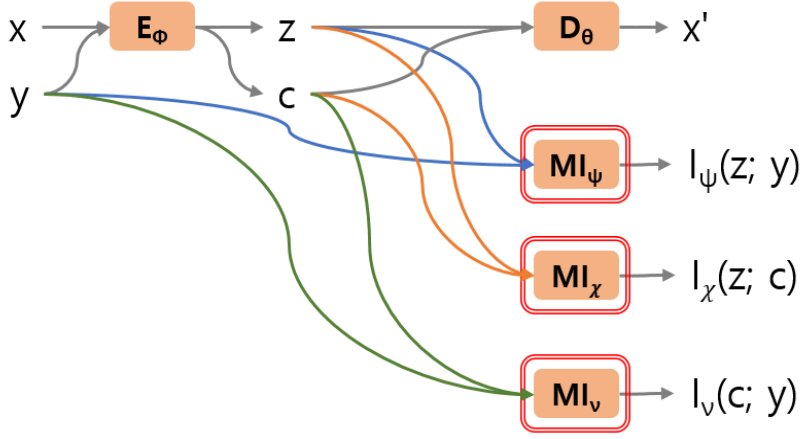


Figure 6. Structure of MMVAE. The model approximates and regularizes mutual information between each pair of three variables of a true attribute ( $y$ ) and latent variables ( $z, c$ ).

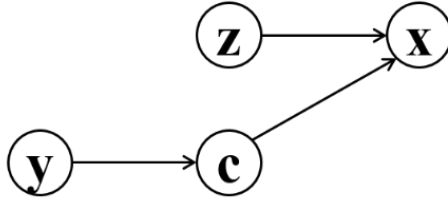


Figure 7. Graphical model of the true generation.  $y$  represents a simple high level attribute and  $c$  represents the detailed low level attribute.

element  $y_i \in \{0, 1\}$ , where  $k$  is the number of attributes of  $x$  we are interested in. A training dataset  $\mathcal{D}$  is  $\{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$  with  $x^{(i)} \in \mathcal{X}$  and  $y^{(i)} \in \mathcal{Y}$ . A whole latent space for representation in

MMVAE is  $\mathcal{Z} \times \mathcal{C}$ . Here  $c_i$  is element of  $c = \{c_1, \dots, c_k\} \in \mathcal{C}$  and usually has 1 or 2 dimensions to contain information associated with only the attribute  $y$ , and  $\mathcal{Z}$  has much more dimensions to contain all the rest of the information.

In our setting, the true generation process of data  $x$  is shown in Figure 7. Therefore joint log-likelihood of  $(x, y, c, z)$  can be decomposed as

$$\begin{aligned} \log p_\theta(x, y, z, c) \\ = \log p_\theta(x|z, c) + \log p(z) + \log p(c|y) + \log p(y). \end{aligned} \quad (3-1)$$

Using a posterior  $q_\phi(z, c|x, y)$  approximated by a encoder in the MMVAE, a lower bound of the model evidence is given by

$$\begin{aligned} \log p_\theta(x, y) \\ = \log \mathbb{E}_{q_\phi(z, c|x, y)} [p_\theta(x, y, z, c)/q_\phi(z, c|x, y)] \\ \geq \mathbb{E}_{q_\phi(z, c|x, y)} [\log p_\theta(x, y, z, c)/q_\phi(z, c|x, y)], \end{aligned} \quad (3-2)$$

in which Jensen's inequality is used because the log function is concave. Then, if  $\log p_\theta(x, y, z, c)$  is substituted by (3-1) the ELBO of MMVAE is represented as follows,

$$\begin{aligned}
ELBO = \mathbb{E}_{q_\phi(z, c|x, y)} [\log p_\theta(x|z, c)] \\
-D_{KL} \left( q_\phi(z|x, y) || p(z) \right) \\
-D_{KL} \left( q_\phi(c|x, y) || p(c|y) \right) + \log p(y),
\end{aligned} \tag{3-3}$$

where  $\log p(y)$  is constant in the dataset, so we can ignore this term when the objective function is optimized.

### 3.2.2 Mutual Information Regularization

As mentioned in section 2.2, Klys et al. (2018) also use a regularizer to reduce mutual information  $I(z;y)$  in CSVAE. They approximate  $I(z;y)$  as follows,

$$\begin{aligned}
I(z;y) &= H(y) - H(y|z) \\
&\approx H(y) + \mathbb{E}_{q_\phi(z|x)\mathcal{D}(x)} \left[ \int_y q_\delta(y|z) \log q_\delta(y|z) dy \right].
\end{aligned} \tag{3-4}$$

In (3-4),  $H(y)$  can be calculated accurately because the value of  $p(y)$  can be obtained almost precisely from the dataset. Also, since the values of label  $y$  is fixed, it is possible to optimize  $q_\delta(y|z)$ . However, attribute representation variable  $c$  doesn't have its fixed distribution or values. Therefore  $I(z;c)$  cannot be approximated in the same way as (3-4). To solve this problem, we borrow the MINE method.

We add three regularizer terms related to mutual information, to the

preceding ELBO term. Each is approximated in MINE-manner and can be a tight lower bound of mutual information using a function  $T$  composed of a proper neural network which is parameterized by  $\nu$ ,  $\psi$ , or  $\chi$  respectively. When a minibatch  $(X, Y)^M = \{x^{(i)}, y^{(i)}\}_{i=1}^M$  is a randomly drawn sample of  $M$  data-points from the training dataset  $\mathcal{D}$ , three regularizers are given by

$$I_\nu(c; y)_M = \mathbb{E}_{\hat{\mathbb{P}}_{cy}^{(M)}}[T_\nu(c, y)] - \log \mathbb{E}_{\hat{\mathbb{P}}_c^{(M)} \otimes \hat{\mathbb{P}}_y^{(M)}}[\exp(T_\nu(c, y))], \quad (3-5)$$

$$I_\psi(z; y)_M = \mathbb{E}_{\hat{\mathbb{P}}_{zy}^{(M)}}[T_\psi(z, y)] - \log \mathbb{E}_{\hat{\mathbb{P}}_z^{(M)} \otimes \hat{\mathbb{P}}_y^{(M)}}[\exp(T_\psi(z, y))], \quad (3-6)$$

$$I_\chi(z; c)_M = \mathbb{E}_{\hat{\mathbb{P}}_{zc}^{(M)}}[T_\chi(z, c)] - \log \mathbb{E}_{\hat{\mathbb{P}}_z^{(M)} \otimes \hat{\mathbb{P}}_c^{(M)}}[\exp(T_\chi(z, c))], \quad (3-7)$$

in which  $\hat{\mathbb{P}}^{(M)}$  is the empirical distribution associated with  $M$  *i.i.d.* samples. The empirical distributions including  $z$  or  $c$  are obtained by sampling from  $q_\phi(z, c | x, y)$ . The products of the empirical marginal can be also obtained by independent sampling from each distribution. Those regularizers have to be maximized by optimizing their parameters because each is the lower bound of mutual information. This maximization of the regularizers should be going along while simultaneously optimizing ELBO so that the encoder learn attribute-factorized representation.

### 3.2.3 Objective Function

The last term in (3-3) which is constant can be ignored in the objective function. And the expectations of the ELBO on  $p(\mathbf{x}, \mathbf{y})$  can be carried out by Monte Carlo estimates on minibatches. Then the complete objective function on a minibatch is given by

$$\begin{aligned}
L^M(\theta, \phi, \nu, \psi, \chi; (\mathbf{X}, \mathbf{Y})^M) \\
&= \mathbb{E}_{(\mathbf{X}, \mathbf{Y})^M}[-ELBO] \\
&\quad - \alpha I_\nu(\mathbf{c}; \mathbf{y})_M + \beta I_\psi(\mathbf{z}; \mathbf{y})_M + \gamma I_\chi(\mathbf{z}; \mathbf{c})_M \\
&= \frac{1}{M} \sum_{i=1}^M \left[ -\mathbb{E}_{q_\phi(\mathbf{z}, \mathbf{c}|\mathbf{x}^{(i)}, \mathbf{y}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}, \mathbf{c})] \right. \\
&\quad + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) || p(\mathbf{z})) \\
&\quad + D_{KL}(q_\phi(\mathbf{c}|\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) || p(\mathbf{c}|\mathbf{y}^{(i)})) \left. \right] \\
&\quad - \alpha I_\nu(\mathbf{c}; \mathbf{y})_M + \beta I_\psi(\mathbf{z}; \mathbf{y})_M + \gamma I_\chi(\mathbf{z}; \mathbf{c})_M,
\end{aligned} \tag{3-8}$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are hyperparameters of positive values and the mutual information terms correspond to the equations from (3-5) to (3-7). Note that the sign in front of  $I_\nu(\mathbf{c}; \mathbf{y})_M$  in (3-8) is minus because we designed the objective function so that the encoder make  $I(\mathbf{c}; \mathbf{y})$  maximized. The objective function is minimized by optimizing the parameters  $\theta$  and  $\phi$  in order to increase the model evidence and

regularize the mutual information. And it also maximized by optimizing the parameters  $\nu$ ,  $\psi$ , and  $\chi$  in order to maximize the lower bound of each mutual information. In other words, that is the following minimax game with the objective function:

$$\min_{\theta, \phi} \max_{\nu, \psi, \chi} L^M(\theta, \phi, \nu, \psi, \chi; (X, Y)^M). \quad (3-9)$$

To train the minimax game including MINE, we need a proper training strategy, which is proposed in the next section.

### 3.3 Implementation

We adopt Gaussian distributions as follows to represent prior and conditional distributions as usually used in VAE-based models:

$$p(z) = \mathcal{N}(z|0, \mathbf{I}), \quad (3-10)$$

$$p(c_i|y_i=j) = \mathcal{N}(c_i|\mu_{i,j}, \text{diag}(\sigma_{i,j}^2)), \quad (3-11)$$

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \text{diag}(\sigma_\phi^2(x))), \quad (3-12)$$

$$q_\phi(c|x, y) = \mathcal{N}(c|\mu_\phi(x, y), \text{diag}(\sigma_\phi^2(x, y))), \quad (3-13)$$

$$p_\theta(x|z, c) = \mathcal{N}(x|\mu_\theta(z, c), \text{diag}(\sigma_\theta^2(z, c))). \quad (3-14)$$

Using the Gaussian forms from (3-10) to (3-13) is convenient to train the



objective function because two KL divergence terms in (3-8) are analytically solved. We choose  $\mu_{i,j}$  and  $\sigma_{i,j}^2$  arbitrarily in (3-11) so that  $c$  is adequately separated by the value of  $y_i$ . A data example  $x$  contains all of the information with respect to an attribute  $y$ , so encoded  $z$  is conditionally independent of  $y$  given  $x$ . For this reason we use  $q_\phi(z|x)$  instead of  $q_\phi(z|x, y)$  to encode  $z$  in (3-12).

The expectation over the product of the marginal distributions in (3-4) to (3-7) is estimated using empirical values which are sampled marginally (not jointly) from each marginal distributions.

To train the minimax game (3-9), we adopt the strategy updating two groups of the parameters alternatively. First group of the parameters is  $\theta$  and  $\phi$  in the encoder and decoder. Second is  $\nu$ ,  $\psi$ , and  $\chi$  in the networks which compute mutual information. The detailed procedure is presented in Algorithm 1.

---

Algorithm 1 Minibatch training of the objective function (3-8) in MMVAE.

---

$\theta, \phi, \nu, \psi, \chi \leftarrow$  Initialize parameters

**for** number of training iterations **do**

$$\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^M \sim \mathcal{D}$$

$$\{\bar{\mathbf{x}}^{(i)}, \bar{\mathbf{y}}^{(i)}\}_{i=1}^M \sim \mathcal{D}$$

$$\mathbf{z}^{(i)}(\phi) \sim q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})$$

$$\mathbf{c}^{(i)}(\phi) \sim q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$$

$$\bar{\mathbf{c}}^{(i)}(\phi) \sim q_{\phi}(\mathbf{z}|\bar{\mathbf{x}}^{(i)}, \bar{\mathbf{y}}^{(i)})$$

$$I_{\nu}(\mathbf{c}; \mathbf{y})_M \leftarrow \frac{1}{M} \sum_{i=1}^M [T_{\nu}(\mathbf{c}^{(i)}(\phi), \mathbf{y}^{(i)})] - \log \left( \frac{1}{M} \sum_{i=1}^M [\exp T_{\nu}(\mathbf{c}^{(i)}(\phi), \bar{\mathbf{y}}^{(i)})] \right)$$

$$I_{\psi}(\mathbf{z}; \mathbf{y})_M \leftarrow \frac{1}{M} \sum_{i=1}^M [T_{\psi}(\mathbf{z}^{(i)}(\phi), \mathbf{y}^{(i)})] - \log \left( \frac{1}{M} \sum_{i=1}^M [\exp T_{\psi}(\mathbf{z}^{(i)}(\phi), \bar{\mathbf{y}}^{(i)})] \right)$$

$$I_{\chi}(\mathbf{z}; \mathbf{c})_M \leftarrow \frac{1}{M} \sum_{i=1}^M [T_{\chi}(\mathbf{z}^{(i)}(\phi), \mathbf{c}^{(i)}(\phi))] - \log \left( \frac{1}{M} \sum_{i=1}^M [\exp T_{\chi}(\mathbf{z}^{(i)}(\phi), \bar{\mathbf{c}}^{(i)}(\phi))] \right)$$

Update parameters by ascending its gradient:

$$\nabla_{\nu, \psi, \chi} [I_{\nu}(\mathbf{c}; \mathbf{y})_M + I_{\psi}(\mathbf{z}; \mathbf{y})_M + I_{\chi}(\mathbf{z}; \mathbf{c})_M]$$

Update parameters by descending its gradient:

$$\nabla_{\theta, \phi} L^M(\theta, \phi, \nu, \psi, \chi; (\mathbf{X}, \mathbf{Y})^M)$$

**end for**

**return**  $\theta, \phi, \nu, \psi, \chi$

---

### 3.4 Evaluation Methods

We consider two methods for quantitatively assessing how well a VAE-based model learns the attribute-factorized representation. Each of them assesses an encoder or decoder, respectively.

Firstly, mutual information between attribute representations (or true attributes) and the other representations used in a model is a good metric to evaluate how well the encoder learned the attribute-factorized representation. MINE allow us to estimate mutual information between variables used in the model after or during training. Even though a testing model using MINE calculates a lower bound of the mutual information, the bound is known to be tight when a sufficiently large size neural network is used in MINE. So, if the bound converges toward zero, we can think that the model has learned the attribute-factorized representation well. It is not the case that we use the same functions in the testing model and the regularizer (3-7) of the objective function. The testing model is completely different function from  $I_X(z; c)$  used in the objective function (3-8) even when the architectures of both neural networks are identical to each other. The testing model is newly trained using the values of the representation variables saved after training the minimax game (3-9), so the values of parameters are entirely different within both neural networks.

Secondly, there is another method to evaluate how well the decoder generates data samples containing given attributes by using the attributes representation variable. This makes use of a classifier which predicts

used attributes from reconstructed samples. The classifier is trained from an original dataset independent of our model. Then classifier learns key features needed to judge attributes from the original dataset. If the decoder can generate the samples that well contain any of attributes we want, then the classification error must be low in the samples generated by the decoder. In fact, the classification error is affected not only by the decoder but also by the encoder. If the encoder cannot learn the attribute-factorized representation well, the decoder using the representation as inputs also cannot learn a proper method to exploit the attribute-factorized representation.

We will use these two methods to assess MMVAE and compare it with previous other works.

## Chapter 4. Experimental Results

In this chapter, we first introduce the experimental setting, such as dataset, network architectures, and optimizers, for implementing MMVAE. Then, we show both qualitative and quantitative evaluation results.

### 4.1 Experimental Setup

#### 4.1.1 Dataset

We used the image dataset of CelebA (Liu et al., 2015), which consist of over 200,000 of face images with 40 different binary-labelled attributes for each image. We cropped the face region into  $138 \times 138$ , and then resized it into  $64 \times 64$  pixels to reduce the required computing resource. As the result, the data domain  $X$  has images with 12,288 dimensions. The attributes used in the experiments are ‘Eyeglasses’ and ‘Male’. We used the whole set of images as the dataset or its subset CelebA-Glasses (39579 images) to mitigate unbalanced portion of the ‘Eyeglasses’ attribute as in Klys et al. (2018).

The model was trained on the 150,000 images of the CelebA or the 30,000 images of the CelebA-Glasses. Half of the remainder was used as the validation dataset and the other half as the test dataset.

### 4.1.2 Architecture of Neural Networks and Training

We used three kinds of neural networks to implement the encoder, decoder and mutual information regularizers, respectively.

Let  $C_n$  be a Convolution-BatchNorm-leakyReLU layer, in which Convolution uses  $n$  filters of size  $4 \times 4$  with a stride of 2 and a padding of 1 (CNN, LeCun, 1989, Ioffe and Szegedy, 2015, Maas et al., 2013, Lample et al., 2017). And let  $FL_n$  be a full connection-BatchNorm-leakyReLU layer with the width of  $n$ . The encoder consists of the following layers:

$$\begin{array}{l} C_{32}-C_{64}-C_{128}-C_{256}-C_{512}-\text{reshaping}-FL_{512\times 2}\\ \quad |\\ \quad C_{256+2k}-C_{256+2k}-\text{reshaping}-FL_{2k\times 2}\end{array}$$

The mean and standard deviation of the latent representation variable  $z$  are calculated as the results of  $FL_{512 \times 2}$ , and those of the attribute representation variable  $c$  are calculated as results of  $FL_{2k \times 2}$ . That is,  $z$  and  $c$  have dimensions of 512 and  $2k$ , respectively. In the  $C_{256+2k}$ , the one-hot vectors ( $[1, 0]$  or  $[0, 1]$ ) representing the true binary attributes  $y$  are concatenated to each hidden layers to encode the attribute information in  $c$ . Here,  $k$  is the number of used attributes.

Now, let  $TC_n$  be a transposed Convolution-BatchNorm-ReLU layer with  $n$  filters, in which transposed Convolution is used for the up-sampling (Zeiler, 2010). And let  $F_n$  be a full connection-BatchNorm-

ReLU layer with a width of  $n$ . The decoder consists of the following layers:

$$F_{512 \times 2 \times 2} - \text{reshaping} - TC_{256+2k} - TC_{128+2k} - TC_{64+2k} - TC_{32+2k}$$

In each  $TC_n$ , the attribute representation variable  $c$  is concatenated to each layers.

For the mutual information regularizers, we used fully connected networks which consist of 7 hidden layers with a width of 512.

The Adam algorithm (Kingma and Ba, 2014) was used as the optimizers during all training. We use an initial learning rate of 0.001 and scheduler with milestones =  $\{2^i | i = 0, 1, 2, \dots\}$  and decay rate = 0.75 for training the encoder and decoder. For training mutual information regularizers, a fixed learning rate of 0.00001 was used.

In *ELBO*, we didn't tune any hyperparameters on two *KL*-divergence terms even though many studies have used them to improve the quality of generated images or regularize the representation space (Klys et al., 2018, Higgins et al., 2017, Chen et al., 2018). This is because the purpose of our study is not to improve the quality of generated images or to confirm the effect of *KL*-divergence terms in VAE but to confirm the effect of regularizing mutual information between each variable. For that reason, only the hyperparameters on the mutual information regularizers were adjusted. Also, in our experimental setting, we have confirmed the attribute representation space is very well divided by the true attribute  $y$  due to the effect of the second *KL*-divergence term in

*ELBO* (3-3) as shown in Figure 8. This means that the mutual information between  $c$  and  $y$  is already at its maximum (same as  $H(y)$ ) even without regularizing  $I_v(c; y)_M$  (3-5). Therefore we have dropped this regularizer term in our experiments. However, if the conditional distribution (3-11) of  $c$  is complicated or the weight hyperparameter for the second *KL*-divergence term in *ELBO* is very small, then the regularizer  $I_v(c; y)_M$  could be useful to maximize the mutual information between  $c$  and  $y$ .

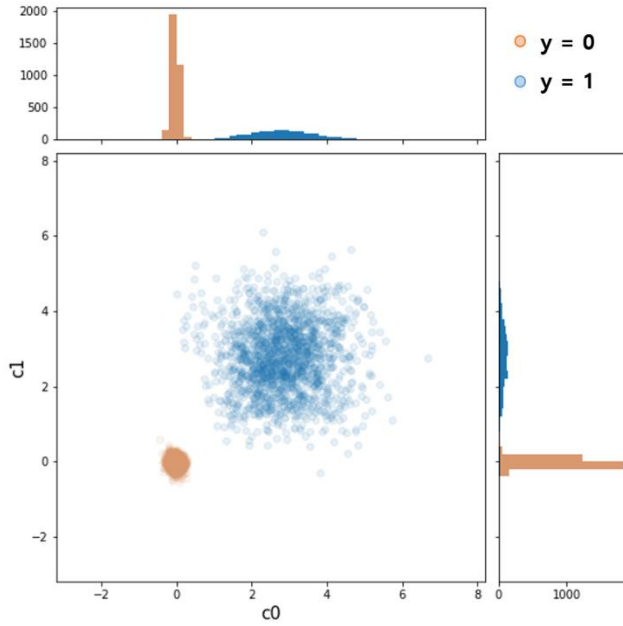


Figure 8. The two dimensional attribute representation space is perfectly divided by the  $y$  value. Here we let  $\mu_{i,0} = (0, 0)$ ,  $\sigma_{i,0} = (0.1, 0.1)$  and  $\mu_{i,1} = (3, 3)$ ,  $\sigma_{i,1} = (1, 1)$ . These are arbitrary to set the conditional distribution (3-11) of attribute representation variable.

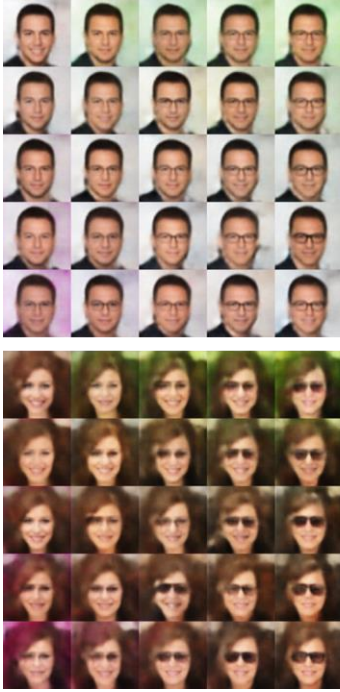


## 4.2 Experimental Results

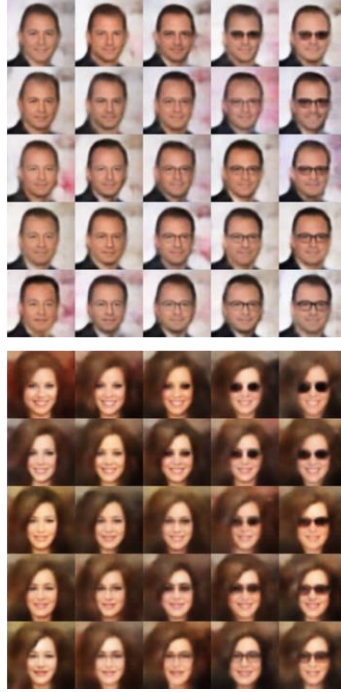
We compare three models including revised-CVAE (r-CVAE), CSVAE, and MMVAE for qualitative and quantitative evaluation. We newly made r-CVAE in which additional attribute representation is included to the original representation in conventional CVAE. r-CVAE has the same representation space as CSVAE and MMVAE but does not have any mutual information regularizer. CSVAE have only one regularizer on  $I(z; y)$ . On the other hand, MMVAE can have three regularizers on  $I(c; y)$ ,  $I(z; y)$  and  $I(z; c)$ .

### 4.2.1 Qualitative Results

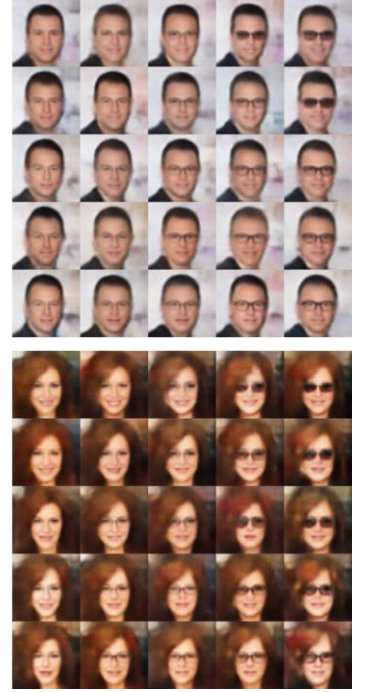
As shown in Figure 9(b) and 9(c), MMVAE, like CSVAE, can manipulate the ‘Eyeglasses’ attribute by changing attribute representation values while fixing other representations. Also, it can represent various sub-attributes of the ‘Eyeglasses’ attribute such as thickness of a frame and lens darkness. It is arbitrary, not specified, which sub-attributes are associated with axes in the attribute representation space. Although various sub-attributes are represented by changing the value in the attribute representation space, the remaining features are almost unchanged in the images. From the results, we can see that the attribute information is effectively separated from other representation information and each information is encoded in different latent variables  $c$  and  $z$ . It is possible by the mutual information



(a) r-CVAE



(b) CSVAE



(c) MMVAE

Figure 9. Generated images on CelebA-Glasses by each of the models trained using the ‘Eyeglasses’ attribute. The locations of the images on a  $5 \times 5$  image-panel correspond to the point set of  $\{(c_0, c_1) | c_0, c_1 \in \{-0.5, 0.5, 1.5, 2.5, 3.5\}\}$  in the attribute representation space (Figure 8). Left-top corresponds to  $(-0.5, -0.5)$ .

regularizers suggested in MMVAE and CSVAE.

On the other hand, r-CVAE is able to represent sub-attributes only to a limited extent. It cannot control frame thickness and lens darkness simultaneously (It control only frame thickness in upper images or only lens darkness in lower images on Figure 9(a)). Furthermore, the attribute

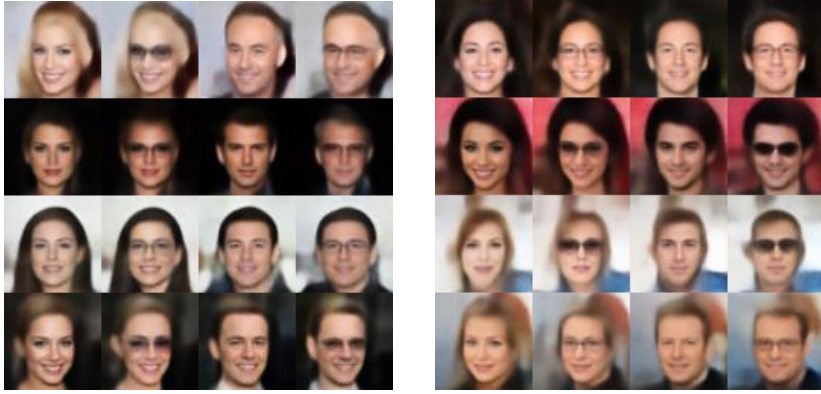


Figure 10. Attribute changed images on CelebA by MMVAE. Used attributes are ‘Male’ and ‘Eyeglasses’.

information cannot be separated from other representation information. Consequently, when the attribute information is changed, color of background is also changed. That's because r-CVAE does not have a mutual information regularizer, so some of information related to the background color is incorrectly encoded in the attribute representation variable  $c$ .

Figure 10 shows the results of manipulating two attributes, ‘Male’ and ‘Eyeglasses’ independently. We train MMVAE using two attributes simultaneously. In each row on image panels, same  $z$  value is used but  $c$  values are randomly selected from the conditional distribution  $p(c|y)$ . Each column corresponds to  $y$  values of  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ , and  $(1, 1)$  for (‘Male’, ‘Eyeglasses’) respectively. Depending on the independent choice of the attributes, the features related to the attribute shown in the images change independently and other features remain intact. This

shows that attribute information is disentangled onto each axis in attribute representation space and also disentangled from other representation information by MMVAE.

### 4.2.2 Quantitative Results

To evaluate how well each VAE-based model factorizes attributes and other representations, we estimate mutual information between attribute representations  $c$  (or true attributes  $y$ ) and other representations  $z$ . Mutual information can be approximated very tightly by MINE. To do this, in each model,  $z$  and  $c$  values are calculated and stored by its encoder during training. After training, new MINE networks are trained from stored  $z$ ,  $c$ , and  $y$  values, and then, the networks can calculate mutual information of each pair among  $z$ ,  $c$  and  $y$  on test dataset. The architectures of the new MINE networks are same to the mutual information regularizers in MMVAE.

First, we evaluate the change in mutual information depending on the hyperparameters  $\beta$  and  $\gamma$  used for the mutual information regularizer  $I_\psi(z; y)_M$  and  $I_\chi(z; c)_M$  in the objective function (3-8) of MMVAE. Another hyperparameter  $\alpha$  for  $I_\nu(c; y)_M$  is set to 0 as mentioned in section 5.1.2. In Figure 11,  $I(c; y)$  has its maximum value  $H(y)(= 0.918 \text{ bits})$  regardless of  $\beta$  and  $\gamma$  because the attribute representation space is already strongly regularized by second  $KL$ -divergence term in ELBO. In contrast,  $I(z; y)$  and  $I(z; c)$  decrease as the hyperparameters increases. By explicitly regularizing each mutual

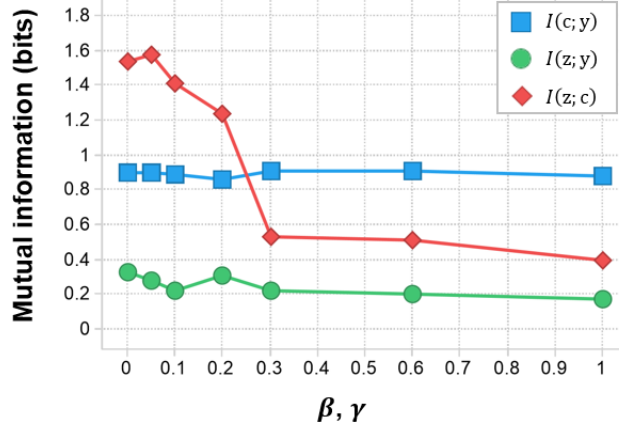


Figure 11. Mutual information depending on the hyperparameters  $\beta$  and  $\gamma$  in MMVAE. To calculate the values of mutual information, a MINE network are trained newly for each data point.

information during training the model, the trained model has lower values of  $I(z; y)$  and  $I(z; c)$ . When the hyperparameters were over 1, we found that the quality of images became degraded because the weights on mutual information regularizers were too much compared to negative log-likelihood in ELBO. In the subsequent results, we used 1 for both of  $\beta$  and  $\gamma$  in MMVAE since we could not find any further improvement in the independent optimization of the two hyperparameters.

Figure 12 shows trends of validation loss and mutual information in the training procedure of the r-CVAE without the MI regularizer.  $I(c; y)$  is rapidly saturated at its maximum value before one epoch. This indicates that the attribute representation is very easy to learn the information of the true attribute label as shown in Figure 8.

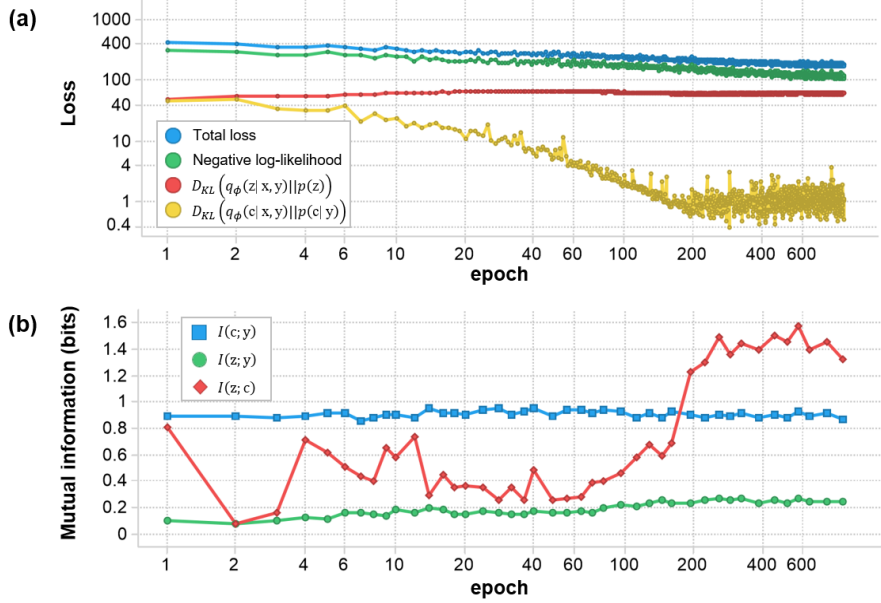


Figure 12. Trends of (a) validation loss and (b) mutual information values during training r-CVAE on CelebA-Glasses.

$D_{KL}(q_\phi(c|x, y)||p(c|y))$  is minimized and  $q_\phi(c|x, y)$  converges to  $p(c|y)$  between epoch of 100 and 200. In that region, the mutual information between  $z$  and  $c$  increases steeply above  $I(c; y)$  (red plot on Figure 12(b)). This means that  $c$  has begun to include additional representation information in addition to attribute information. For that reason, when generating images using r-CVAE with fixed  $z$  and changing  $c$ , some features unrelated to the attribute are changed as well (Figure 9(a)). However, MMVAE and CSVAE can suppress the increase of  $I(z; c)$  by using mutual information regularizers as shown in Figure 13. In particular, the value of  $I(z; c)$  is lower than that of CSVAE because

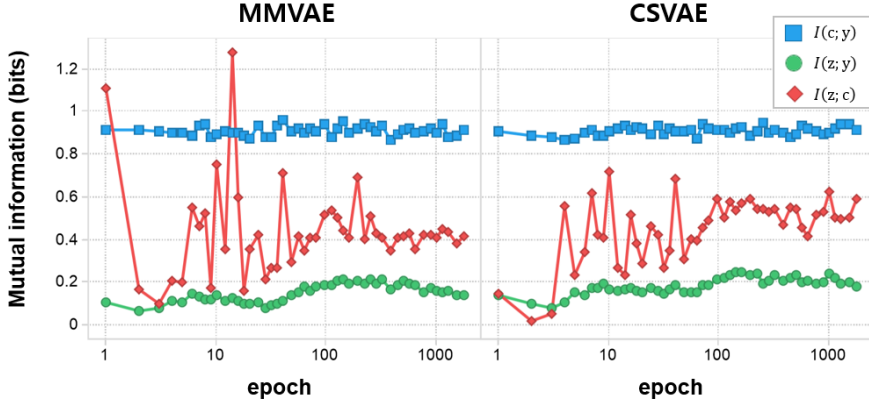


Figure 13. Trends of mutual information values during training MMVAE and CSVAE on CelebA-Glasses.

MMVAE regularizes  $I(z; c)$  to be minimized explicitly unlike CSVAE in which the regularization is impossible due to its method approximating mutual information. The values of mutual information after training of 1000 epochs of three models are shown in Table 2. The values of  $I(c; y)$  are close to its maximum in all models by the second  $KL$ -divergence term in their ELBO (3-3), but  $I(z; y)$  and  $I(z; c)$  have the lowest values in MMVAE. It confirms that MMVAE best learns the attribute-factorized representations.

We confirm how well each model represent desired attributes in images when it reconstruct the images. To do this we use an attribute classifier using a convolutional network pre-trained on original dataset. Accuracy of the classifier is 98.46 percent on original test dataset. The images to be tested are generated by swapping the attribute-representation from original images in each model. First, the

Table 2. Mutual information between each pair among  $z$ ,  $c$ , and  $y$  on each model. Each is the mean $\pm$ standard deviation of five values approximated by MINE method.

	Mutual information (bits)		
	$I(c; y)$	$I(z; y)$	$I(z; c)$
<b>r-CVAE</b>	0.895 $\pm$ 0.014	0.245 $\pm$ 0.005	1.391 $\pm$ 0.044
<b>CSVAE</b>	0.909 $\pm$ 0.012	0.210 $\pm$ 0.018	0.498 $\pm$ 0.051
<b>MMVAE</b>	0.908 $\pm$ 0.016	<b>0.175<math>\pm</math>0.017</b>	<b>0.400<math>\pm</math>0.025</b>

representations of  $z$  and  $c$  is extracted from original images by the encoder of each model, and the values of  $c$  are replaced with the values sampled from  $p(c|y)$  given the attributes opposite to the original. The decoder then reconstructs the images containing the opposite attribute. We classify reconstructed images whose the attributes have been swapped by each model, using the pre-trained classifier. The results are shown in Table 3. Since MMVAE has learned to distinguish between attribute representations and other representations, the model represents desired attributes well on the images by changing only attribute representation values when reconstructing images. If the attribute-factorized representation is not well learned in a model, attribute information of the original images remains in  $z$ , so that the desired attribute may not be represented well in the generated images even if the attribute representations values are changed.



Table 3. Accuracy of classification of images reconstructed by each model using the swapped attribute of ‘Eyeglasses’ on CelebA-Glasses.

	<b>r-CVAE</b>	<b>CSVAE</b>	<b>MMVAE</b>
Accuracy	94.35%	96.50%	<b>97.17%</b>

## Chapter 5. Conclusion

### 5.1 Conclusions

We have designed a new VAE-based model for generating images with the given attributes, where separating attribute information and the rest of the information is achieved by explicitly enforcing mutual information conditions between the two information sets to be minimized. In particular, explicit regularization for the separation of these two sets of information in representation space was not possible in the previous works, but we made it possible by adopting MINE in our model. We confirmed through the experiments on CelebA dataset that the encoder of the model learns the attribute representation and the other representation in a better factorized form, by demonstrating that  $I(z; y)$  and  $I(z; c)$  in the model have smaller values than those in r-CVAE and CSVAE. We also showed, through showing generated images and classification results, that the given attributes are better represented in images when the images are generated by the decoder of the model using the learned attribute-factorized representation space. Even though MMVAE has been demonstrated only in the task of generating images, it can extend the application range to the other domains such as translation, speech generation, composition and so on, using the core concept of MMVAE.

## 5.2 Contributions

The contributions of this thesis can be summarized as below.

- We constructed MMVAE and its objective function, which can better learn attribute-factorized representations by explicitly regularizing mutual information between attributes representation and the other representation variables.
- We established a proper strategy to train VAE-based generative models while simultaneously calculating a tight lower bound of mutual information by adopting MINE. It differs from the previous works in that it can approximate the mutual information even if marginal distributions of variables are unknown in a VAE-based model.
- We confirmed the trends of mutual information during training CVAE-based models. Without a mutual information regularizer, the mutual information between representations increases sharply after  $KL$ -divergence converges.

## 5.3 Limitations

MMVAE uses three mutual information regularizer terms. From

this, there are two disadvantages even though MMVAE is successful. First, each mutual information term is composed of its own neural network to calculate a lower bound of mutual information as tightly as possible. Therefore more computing resources are needed to train the model. Second disadvantage is related to the hyperparameter optimization due to the extra hyperparameters for mutual information regularizers. If we want to increase the quality of the image or want to shape the representation space better into prior distribution, the number of hyperparameters increases to the number of weights terms of the ELBO (3-3). It might be helpful to use proper hyperparameter optimization strategies such as Spearmint (Snoek et al., 2012), TPE (Bergstra et al., 2011), and SMAC (Hutter et al., 2011), but another problem is that there is no integrated metric to evaluate the performance of the VAE-based models. The evaluation of the quality of the images produced is subjective, either personally or through the Mechanical Turk platform (Lample et al., 2017, Salimans et al., 2016, Wang, et al., 2018), and these evaluations cannot be used for the aforementioned hyperparameter optimization strategies. There is also no metric to evaluate generative models of whether a distribution over reconstructed samples can be well generated similar to the original data distribution having various modes. Therefore, many studies related to generative models have made use of arbitrary metrics that show only the specific utility of their models.

Another limitation of our works is related to quality of generated samples, especially images. Even if MMVAE can ideally learn the

attribute-factorized representation, that is irrelevant to generating good images like the original. To improve the quality of the generated images in VAE-based models, order strategies are needed such as modifying networks or using a discriminator like VAE/GAN (Larsen et al., 2016).

In addition, MMVAE is limited to a supervised method that uses labels included in dataset. Therefore, attributes whose labels are not included in the dataset cannot be manipulated in the model. There are several generative models manipulating attribute representations even if those are unsupervised models and don't use any labels, as mentioned in the end of section 2.2. These models have the potential to learn and manipulate the attributes representation without labels. However there is a difference from MMVAE in that it is not guaranteed to learn the specific attribute we want.

## References

- Bao, J., Chen, D., Wen, F., Li, H., & Hua, G. (2017). CVAE-GAN: fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2745-2754).
- Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Hjelm, D., & Courville, A. (2018, July). Mutual Information Neural Estimation. In International Conference on Machine Learning (pp. 530-539).
- Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Advances in neural information processing systems (pp. 2546-2554).
- Chen, T. Q., Li, X., Grosse, R. B., & Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In Advances in Neural Information Processing Systems (pp. 2610-2620).
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems (pp. 2172-2180).
- Cover, T. M., & Thomas, J. A. (2012). Elements of information theory. John Wiley & Sons.
- Creswell, A., Mohamied, Y., Sengupta, B., & Bharath, A. A. (2017). Adversarial Information Factorization. arXiv preprint arXiv:1711.05175.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), 303-314.
- Donsker, M. D., & Varadhan, S. S. (1983). Asymptotic evaluation of certain

Markov process expectations for large time. IV. Communications on Pure and Applied Mathematics, 36(2), 183-212.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., ... & Meent, J. W. (2019, April). Structured disentangled representations. In The 22nd International Conference on Artificial Intelligence and Statistics (pp. 2525-2534).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., ... & Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations (Vol. 3).

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

Huang, X., & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1501-1510).

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011, January). Sequential model-based optimization for general algorithm configuration. In International Conference on Learning and Intelligent Optimization (pp. 507-523). Springer, Berlin, Heidelberg.

Ioffe, S., & Szegedy, C. (2015, June). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In International Conference on Machine Learning (pp. 448-456).

Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for

- generative adversarial networks. arXiv preprint arXiv:1812.04948.
- Kim, H., & Mnih, A. (2018). Disentangling by factorising. arXiv preprint arXiv:1802.05983.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
- Klys, J., Snell, J., & Zemel, R. (2018). Learning Latent Subspaces in Variational Autoencoders. In *Advances in Neural Information Processing Systems* (pp. 6445-6455).
- Kolchinsky, A., Tracey, B. D., & Wolpert, D. H. (2017). Nonlinear information bottleneck. arXiv preprint arXiv:1705.02436.
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6), 066138.
- Lample, G., Zeghidour, N., Usunier, N., Bordes, A., & Denoyer, L. (2017). Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems* (pp. 5967-5976).
- Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *33rd International Conference on Machine Learning (ICML 2016) International Conference on Machine Learning*.
- LeCun, Y. (1989). Generalization and network design strategies. In *Connectionism in perspective* (Vol. 19). Amsterdam: Elsevier.
- Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can



- approximate any function. *Neural networks*, 6(6), 861-867.
- Li, C., & Wand, M. (2016, October). Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision* (pp. 702-716). Springer, Cham.
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision* (pp. 3730-3738).
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, No. 1, p. 3).
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234-2242).
- Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951-2959).
- Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems* (pp. 3483-3491).
- Taigman, Y., Polyak, A., & Wolf, L. (2016). Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

- Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8798-8807).
- Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010, June). Deconvolutional networks. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 2528-2535). IEEE.

## 초록

최근, 데이터의 표현 (representation)을 학습하고 새로운 샘플을 생성 할 수 있는 심층 생성 모델에 대한 연구가 활발하다. 우리는 특정한 속성 (attribute) 및 다른 속성과 관련된 표현들의 관계를 고려하여, 이들을 심층 생성 모델에서 어떻게 구분하여 처리할지에 대해 고찰하였다. 본 연구에서는 다수의 상호 정보량 (mutual information) 성분을 정규화하여 표현에서 속성의 요소분리를 강화시킬 수 있는, 변분법적 오토인코더 (Variational Autoencoder, VAE) 기반의 새로운 생성 모델 (MMVAE : Multiple mutual information VAE)과 목적 함수를 소개한다. 특히 Mutual Information Neural Estimation (MINE, Belghazi et al., 2018)을 채택하여, 속성의 레이블, 속성 표현 및 다른 표현 사이의 상호 정보량을 명시적으로 정규화하는 프레임 워크를 구성하였다. 이 모델에서 목적 함수는 증거 하한값 (evidence lower bound, ELBO)과 세 개의 상호 정보량으로 구성된다. 이는 미니맥스 게임 (mini-max game)에 해당하는데, 오토인코더의 매개 변수 그룹은 목적 함수를

최소화하도록 최적화되지만 상호 정보량의 매개 변수 그룹은 목적 함수를 최대화하도록 최적화 된다. 우리는 CelebA 데이터 세트에 대한 일련의 실험을 통해, MMVAE가 속성 표현과 다른 표현을 더 잘 구분하여 학습할 수 있고 이렇게 학습된 속성-요소분리된 표현 (attribute-factorized representation)은 주어진 속성을 포함하는 이미지를 생성하는 데 유용하다는 것을 입증하였다.

**Keywords:** 심층 생성 모델, 표현 학습, 속성-요소분리, 상호정보량 정규화

**Student Number:** 2017-21650